STRUCTURE

STRUCTURE IN INPUTS

STRUCTURE IN OUTPUTS

STRUCTURE IN META-LEARNING MODEL
THIS TALK
Structure & Meta-learning
STATISTICAL RELATIONAL LEARNING

1. Make use of logical structure
2. Handle uncertainty
3. Perform collective inference

[GETOOR & TASKAR '07]
PROBABILISTIC SOFT LOGIC (PSL)

A probabilistic programming language for collective inference problems

- Predicate = relationship or property
- Ground Atom = (continuous) random variable
- Weighted Rules = capture dependency or constraint

PSL Program = Rules + Input DB

psl.linqs.org

COLLECTIVE Reasoning
outputs depend on each other
COLLECTIVE Classification Pattern

local-predictor(x, l) \rightarrow label(x, l)
label(x, l) & link(x, y) \rightarrow label(y, l)
COLLECTIVE Classification Pattern

local-predictor(x,1) \rightarrow label(x,1)

label(x,1) \& link(x,y) \rightarrow label(y,1)
COLLECTIVE CLASSIFICATION

QUESTION: or ?
COLLECTIVE CLASSIFICATION

QUESTION: or ?
COLLECTIVE CLASSIFICATION

QUESTION: or ?
COLLECTIVE CLASSIFICATION

Local rules:
- “If X donates to party P, X votes for P”
- “If X tweets party P slogans, X votes for P”

Relational rules:
- “If X is linked to Y, and X votes for P, Y votes for P”
COLLECTIVE CLASSIFICATION

Local rules:
• “If X donates to party P, X votes for P”
• “If X tweets party P slogans, X votes for P”

Relational rules:
• “If X is linked to Y, and X votes for P, Y votes for P”

Donates(X,P) $\rightarrow$ Votes(X,P)
Local rules:
- “If X donates to party P, X votes for P”
- “If X tweets party P slogans, X votes for P”

Relational rules:
- “If X is linked to Y, and X votes for P, Y votes for P”

\[
\text{Tweets}(X, \text{“Affordable Health”}) \Rightarrow \text{Votes}(X, \text{“Democrat”})
\]
COLLECTIVE CLASSIFICATION

Local rules:
- “If X donates to party P, X votes for P”
- “If X tweets party P slogans, X votes for P”

Relational rules:
- “If X is linked to Y, and X votes for P, Y votes for P”

$\text{Votes}(X, P) \land \text{Friends}(X, Y) \Rightarrow \text{Votes}(Y, P)$

$\text{Votes}(X, P) \land \text{Spouse}(X, Y) \Rightarrow \text{Votes}(Y, P)$
COLLECTIVE Activity Recognition

inferring activities in video sequence
ACTIVITY RECOGNITION

crossing  waiting  queueing  walking  talking  dancing  jogging
COLLECTIVE Pattern

- local-predictor(x,l,f) → activity(x,l,f)
- activity(x,l,f) & same-frame(x,y,f) → activity(y,l,f)
- activity(x,l,f) & next-frame(f,f') → activity(x,l,f')
EMPIRICAL HIGHLIGHTS

Improved activity recognition in video:

<table>
<thead>
<tr>
<th></th>
<th>5 Activities</th>
<th>6 Activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOG</td>
<td>47.4%</td>
<td>.481 F1</td>
</tr>
<tr>
<td>HOG + PSL</td>
<td>59.8%</td>
<td>.603 F1</td>
</tr>
<tr>
<td>ACD</td>
<td>67.5%</td>
<td>.678 F1</td>
</tr>
<tr>
<td>ACD + PSL</td>
<td>69.2%</td>
<td>.693 F1</td>
</tr>
</tbody>
</table>

COLLECTIVE Stance Prediction

Inferring users’ stance in online debates
DEBATE STANCE CLASSIFICATION

TOPIC: Climate Change

TASK:
Jointly infer users’ attitude on topics and interaction polarity

Sridhar, Foulds, Huang, Getoor & Walker, Joint Models of Disagreement and Stance, ACL 2015
// local text classifiers
w_1: LocalPro(U, T) -> Pro(U, T)
w_1: LocalDisagree(U_1, U_2) -> Disagrees(U_1, U_2)

// Rules for stance
w_2: Pro(U_1, T) & Disagrees(U_1, U_2) -> !Pro(U_2, T)
w_2: Pro(U_1, T) & !Disagrees(U_1, U_2) -> Pro(U_2, T)

// Rules for disagreement
w_3: Pro(U_1, T) & Pro(U_1, T) -> !Disagrees(U_1, U_2)
w_3: !Pro(U_1, T) & Pro(U_2, T) -> Disagrees(U_1, U_2)

bitbucket.org/linqs/psl-joint-stance
## PREDICTING STANCE IN ONLINE FORUMS

**Task:** Predict post and user stance from two online debate forums

- 4Forums.com: ~300 users, ~6000 posts
- CreateDebate.org: ~300 users, ~1200 posts

### 4FORUMS.COM

<table>
<thead>
<tr>
<th></th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text-only Baseline</td>
<td>69.0</td>
</tr>
<tr>
<td>PSL</td>
<td>80.3</td>
</tr>
</tbody>
</table>

### CREATEDEBATE.ORG

<table>
<thead>
<tr>
<th></th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text-only Baseline</td>
<td>62.7</td>
</tr>
<tr>
<td>PSL</td>
<td>72.7</td>
</tr>
</tbody>
</table>

LINK Prediction Pattern

\[ \text{link}(x, y) \land \text{similar}(y, z) \rightarrow \text{link}(x, z) \]
CLUSTERING
Pattern

link(x, y) & link(y, z) → link(x, z)
MATCHING Pattern

\[ \text{link}(x,y) \land \neg \text{same}(y,z) \rightarrow \neg \text{link}(x,z) \]
SRL <-> META-LEARN

SRL Concepts
- Templated Models
- Weight Learning
- Structure Learning
- Latent Variables
- Logical rules

Meta-learning Concepts
- Tied Hyperparameters
- Hyperparameter Optimization
- Feature & Algorithm Selection
- Landmarks
- Few/Zero-shot learning
Probabilistic programming language for defining distributions

/* Local rules */
\[ w_d : \text{Donates}(A, P) \rightarrow \text{Votes}(A, P) \]
\[ w_t : \text{Mentions}(A, "Affordable Health") \rightarrow \text{Votes}(A, "Democrat") \]
\[ w_t : \text{Mentions}(A, "Tax Cuts") \rightarrow \text{Votes}(A, "Republican") \]

/* Relational rules */
\[ w_s : \text{Votes}(A, P) \& \text{Spouse}(B, A) \rightarrow \text{Votes}(B, P) \]
\[ w_f : \text{Votes}(A, P) \& \text{Friend}(B, A) \rightarrow \text{Votes}(B, P) \]
\[ w_c : \text{Votes}(A, P) \& \text{Colleague}(B, A) \rightarrow \text{Votes}(B, P) \]

/* Range constraint */
\[ \text{Votes}(A, "Republican") + \text{Votes}(A, "Democrat") = 1.0 \]
LEARN when structural patterns hold across many instantiations
STRUCTURE LEARNING

- Large subfield of statistical relational learning
  - Friedman et al. IJCAI 99, Getoor et al. JMLR 02, Kok & Domingos ICML05, Mihalkova & Mooney ICML07, DeRaedt et al. MLJ 2008, Khosravi et al AAAI10, Khot et al. ICDM 11, Van Haaren et al. MLJ15, among others
  - NIPS Relational Representation Learning Workshop

- Basic Idea
  - Search model space
  - Model space is very rich
  - Optimize parameters
    - Information theoretic criteria, likelihood-based, and Bayesian approaches
when structural patterns hold across many learning tasks
META LEARNING

Configurations  Works  Tasks
META LEARNING

Rules express:
• “If configuration C works well for task T1, and task T2 is similar to T1, C will work well for T2”
• “If configuration C1 works well for task T, and configuration C2 similar to C1, C2 will work well for T”
META LEARNING

Rules express:

• “If configuration C works well for task T1, and task T2 is similar to T1, C will work well for T2”
• “If configuration C1 works well for task T, and configuration C2 similar to C1, C2 will work well for T”

\[
\text{Works}(C,T1) \land \text{SimilarTask}(T1,T2) \rightarrow \text{Works}(C,T2)
\]
META LEARNING

Rules express:
• “If configuration C works well for task T1, and task T2 is similar to T1, C will work well for T2”
• “If configuration C1 works well for task T, and configuration C2 similar to C1, C2 will work well for T”

\[
\text{Works}(C1,T) \land \text{SimilarConfig}(C1,C2) \Rightarrow \text{Works}(C2,T)
\]
META-LEARNING

• Challenge: defining similarity

• Advantages:
  • can make use of multiple similarity measures
  • can use domain knowledge for defining task and configuration similarity

• Research questions:
  • Are there benefits from using this approach?
  • What are opportunities for collective reasoning?
LANDMARKING

• Can be described using latent variables

• E.g., Task-Area and Learner-Expertise as latent variables

• Research questions:
  • Are there benefits from using SRL approach?
  • What are opportunities for collective reasoning?
ALGORITHM & MODEL SELECTION

• Can be described using (probabilistic/soft) logical rules

• Research questions:
  • Are there benefits from using SRL approach?
  • What are opportunities for collective reasoning?
PIPELINE CONSTRUCTION

• Can be described using logical rules and constraints

• Research questions:
  • Are there benefits from using SRL approach?
  • What are opportunities for collective reasoning?
CLOSING
STRUCTURE AND META-LEARNING

CLOSING THE LOOP
CLOSING COMMENTS

Provided some examples of structure and collective reasoning

Opportunity for Meta-Learning methods that can mix:
- probabilistic & logical inference
- data-driven & knowledge-driven modeling
- Meta-modeling for meta-modeling

Compelling applications abound!
THANK YOU!

psl.linqs.org

Contact information:
getoor@ucsc.edu

| @lgetoor
PSL SUMMARY IN A SLIDE

- MAP Inference in PSL translates into convex optimization problem → inference is really fast
- Inference further enhanced with state-of-the-art optimization and distributed graph processing paradigms → inference even faster
- Learning methods for rule weights & latent variables
- PSL is open-source, code, data, tutorials available online

psl.linqs.org